

## **AMENDMENTS TO THE CLAIMS**

This listing of claims will replace all prior versions, and listings, of claims in the application:

### **Listing of Claims:**

- 1           1. (Currently amended) A method for managing object to relational one-to-  
2   many mapping for an object model mapped to a relational database, the method  
3   comprising steps of:  
4           obtaining, for a source object having a primary key value and being  
5   manipulated in a corresponding source table of the relational database, mapping  
6   meta-data including information of a corresponding target table for storing at least  
7   one target object with which the source object has a one-to-many relationship of  
8   privately owned type and information of a foreign key of the corresponding target  
9   table;  
10          generating an instruction to manipulate the at least one target object in the  
11   corresponding target table based on the mapping meta-data; and  
12          manipulating the at least one target object in the database by executing the  
13   instruction on the database, wherein the at least one target object is manipulated  
14   by using the foreign key together with the target object data, thereby avoiding  
15   back references in the target objects.
  
- 1           2. (Original) The method as claimed in claim 1, wherein the instruction  
2   generating step generates an instruction to insert, read, delete or update the each  
3   target object in the database.

1           3. (Original) The method as claimed in claim 1, wherein each target object  
2 in the object model does not contain information regarding the one-to-many  
3 relationship.

1           4. (Original) The method as claimed in claim 1, wherein the generating  
2 step and the manipulating step are performed on each target object with which the  
3 source object has the one-to-many relationship of the privately owned type.

1           5. (Original) The method as claimed in claim 1, wherein the obtaining step  
2 is carried out when a system performing the method is initialized.

1           6. (Currently amended) A method for managing object to relational one-to-  
2 many mapping for an object model mapped to a relational database, the method  
3 comprising steps of:

4           obtaining, for a source object having a primary key value and being  
5 inserted in a corresponding source table of the relational database, mapping meta-  
6 data including information of a corresponding target table for storing at least one  
7 target object with which the source object has a one-to-many relationship of  
8 privately owned type and information of a foreign key of the corresponding target  
9 table;

10          generating an insert instruction to add a value of the foreign key based on  
11 the primary key value of the source object, and to insert the at least one target  
12 object into the corresponding target table based on the mapping meta-data; and

13          inserting the at least one target object into the database by executing the  
14 insert instruction on the database, wherein the at least one target object is inserted  
15 by populating the foreign key together with the target object data, thereby  
16 avoiding back references in the target objects.

1           7. (Original) The method as claimed in claim 6, wherein the insert  
2 instruction generating step and the target object inserting step comprise steps of:  
3           a)     building a database row representation of each target object  
4 containing target object data;  
5           b)     adding the primary key value of the source object to the database  
6 row representation;  
7           c)     generating the insert instruction based on the database row  
8 representation and the target object data using the mapping meta-data;  
9           d)     writing the target object in a row of the corresponding target table  
10 by executing the insert instruction on the database; and  
11           e)     repeating steps a) to d) for each target object using the mapping  
12 meta-data.

1           8. (Original) The method as claimed in claim 7, wherein the adding step  
2 adds the primary key value of the source object as a foreign key value of the target  
3 object.

1           9. (Original) The method as claimed in claim 6, wherein the statement  
2 generating step generates the insert instructions as a Structured Query Language  
3 (SQL) Insert statement.

1           10. (Original) The method as claimed in claim 6 further comprising a step  
2 of:  
3           storing the mapping meta-data external to the source object class and the  
4 target object classes.

1           11. (Original) The method as claimed in claim 10, wherein the storing step  
2 stores the mapping meta-data as XML files.

1           12. (Original) The method as claimed in claim 6 further comprising steps  
2 of:  
3           reading the source object from the object model; and  
4           inserting the source object into the corresponding source table in the  
5 database.

1           13. (Currently amended) A method for managing object to relational one-  
2 to-many mapping for an object model mapped to a relational database, the method  
3 comprising steps of:  
4           obtaining, when a source object having a primary key value is being read  
5 from a source table in the relational database, mapping meta-data including  
6 information of one or more corresponding target tables and information of one or  
7 more foreign keys of the corresponding target tables;  
8           generating a select instruction to select from the target tables target objects  
9 with which the source object has one-to-many relationships of privately owned  
10 type, based on the mapping meta-data and the primary key value of the source  
11 object; and  
12           reading the target objects and relationships relating to the source object  
13 from the database by executing the select instruction on the database, wherein the  
14 target objects are read by using the foreign key together with the target object  
15 data, thereby avoiding back references in the target objects.

1           14. (Original) The method as claimed in claim 13, wherein the reading  
2 step comprises steps of:  
3           querying for rows in the target tables that have a foreign key value  
4 matching the primary key value of the source object by executing the select  
5 instruction on the database;

6 translating the queried rows into target objects based on the mapping  
7 meta-data;  
8 adding the target objects to a collection that represents a value of  
9 relationship of the source object, the value of relationship referencing to the target  
10 objects; and  
11 setting the value of the relationship into the source object.

1 15. (Original) The method as claimed in claim 13, wherein the generating  
2 step comprises a step of generating a select instruction to check foreign key fields  
3 in the target tables.

1 16. (Original) The method as claimed in claim 13, wherein the statement  
2 generating step generates the read instructions as SQL Select statements.

1 17. (Original) The method as claimed in claim 16 further comprising a  
2 step of:  
3 storing the mapping meta-data external to the source object class and the  
4 target object classes.

1 18. (Original) The method as claimed in claim 17, wherein the storing step  
2 stores the mapping meta-data as XML files.

1 19. (Currently amended) A method for managing object to relational one-  
2 to-many mapping for an object model mapped to a relational database, the method  
3 comprising steps of:  
4 obtaining, when a source object having a primary key value is being  
5 deleted from a source table in the relational database; mapping meta-data that  
6 defines one or more corresponding target tables storing target objects with which

7 the source object has one-to-many relationships of privately owned type and  
8 foreign key information;  
9 generating a delete instruction to delete the target objects from the target  
10 tables based on the mapping meta-data; and  
11 deleting the target objects by executing the delete instruction on the  
12 database, wherein the target objects are deleted by using the foreign key together  
13 with the target object data, thereby avoiding back references in the target objects.

1 20. (Original) The method as claimed in claim 19 further comprising a  
2 step of determining whether the target objects are stored in a single target table,  
3 and  
4 wherein, when the target objects are determined to be stored in a single  
5 target table, the delete instruction generating step comprises a step of generating a  
6 single delete instruction that deletes all rows from the target table that have a  
7 foreign key matching the primary key of the source object.

1 21. (Original) The method as claimed in claim 19 further comprising a  
2 step of determining whether the target objects are stored in multiple target tables;  
3 and  
4 wherein, when the target objects are determined to be stored in multiple  
5 target tables, the delete instruction generating step comprises steps of:  
6 extracting a primary key value for each target object;  
7 generating a delete instruction to delete each row with the primary key  
8 value; and  
9 repeating the extracting step and the generating step.

1 22. (Original) The method as claimed in claim 19, wherein the statement  
2 generating step generates the delete instructions as SQL Delete statements.

1           23. (Original) The method as claimed in claim 19 further comprising a  
2 step of:  
3           storing the mapping meta-data external to the source object class and the  
4 target object classes.

1           24. (Original) The method as claimed in claim 23, wherein the storing step  
2 stores the mapping meta-data as XML files.

1           25. (Currently amended) A method for managing object to relational one-  
2 to-many mapping for an object model mapped to a relational database, the method  
3 comprising a step of tracking changes to relationships and target objects with  
4 which the source object has one-to-many relationships of privately owned type,  
5 wherein the tracking step comprises steps of:  
6           creating, at the start of a transaction, a source clone of a source object  
7 having a primary key value, a relationship clone of a relationship in the source  
8 object, and a target clone of target objects referenced by the relationship; and  
9           determining, when the transaction is committed, changes to the source  
10 object and target objects by comparing the source clone to the current state of the  
11 source object, and comparing the target clone to the current state of the target  
12 objects using the primary key value of the source object added to the target  
13 objects, wherein the changes to the source object and target objects are determined  
14 by using the foreign key together with the target object data, thereby avoiding  
15 back references in the target objects.

1           26. (Original) The method as claimed in claim 25 further comprising a  
2 step of updating relationship of the source object referencing target objects to the  
3 database wherein the updating step comprises steps of:

4           determining which objects have been added, removed or changed in the  
5 relationship using mapping meta-data including information of corresponding  
6 source table, one or more target tables and foreign keys of the target tables;  
7           generating an insert instruction for each target object of a source target that  
8 has been added;  
9           generating a delete instruction for each target object of a source target that  
10 has been removed; and  
11           generating an update instruction for each target object of a source target  
12 that has been changed.

1           27. (Original) The method as claimed in claim 25, wherein the steps of  
2 generating an insert, delete and update instructions comprise steps of:  
3           generating SQL Insert statements for objects that have been added;  
4           generating SQL Delete statements for objects that have been removed; and  
5           generating SQL Update statements for objects that have been changed.

1           28. (Original) The method as claimed in claim 25 further comprising a  
2 step of:  
3           storing the mapping meta-data external to the source object class and the  
4 target object classes.

1           29. (Original) The method as claimed in claim 28, wherein the storing step  
2 stores the mapping meta-data as XML files.

1           30. (Currently amended) A one-to-many mapping manager for managing  
2 object to relational one-to-many mapping for an object model mapped to a  
3 relational database, the system comprising:  
4           a meta-data receiver for obtaining, for a source data having a primary key  
5 value and being manipulated, mapping meta-data including information of one or



6 more corresponding target tables for storing multiple target objects with which the  
7 source object has one-to-many relationships of privately owned type and  
8 information of one or more foreign keys of the corresponding target tables;  
9 an instruction generator for generating instructions to manage the multiple  
10 target objects and the relationship based on the mapping meta-data; and  
11 an instruction executor for executing the instructions on the relational  
12 database and managing the target objects and the relationship in the database in  
13 accordance with the manipulation of the source object;  
14 wherein by using the foreign key together with the target object data, the  
15 one-to-many mapping manager avoids back references in the target objects.

1 31. (Original) The manager as claimed in claim 30, wherein the instruction  
2 generator having an insert instruction generator of generating insert instructions to  
3 insert the multiple target objects into the corresponding target tables based on the  
4 mapping meta-data when a source object is inserted into the database.

1 32. (Original) The manager as claimed in claim 31, wherein the insert  
2 instruction generating function includes:  
3 a representation builder for building a database row representation for each  
4 target object; and  
5 a primary key information adder for adding the primary key value of the  
6 source object to the database row representation.

1 33. (Original) The manager as claimed in claim 31, wherein the instruction  
2 generator generates the insert instructions as SQL Insert statements.

1 34. (Original) The manager as claimed in claim 30, wherein the instruction  
2 generator having a read instruction generator for generating read instructions to  
3 read the multiple target objects and the relationship from the corresponding target

4 tables and the foreign key values based on the mapping meta-data when a source  
5 object is read from the database.

1 35. (Original) The manager as claimed in claim 34, wherein the instruction  
2 generator generates the read instructions as SQL Select statements.

1 36. (Original) The manager as claimed in claim 30, wherein the instruction  
2 generator having a delete instruction generator for generating delete instructions to  
3 read the multiple target objects and the relationship from the corresponding target  
4 tables and the foreign key values based on the mapping meta-data when a source  
5 object is deleted from the database.

1 37. (Original) The manager as claimed in claim 36, wherein the instruction  
2 generator generates the delete instructions as SQL Delete statements.

1 38. (Original) The manager as claimed in claim 30, wherein the instruction  
2 generator having an update instruction generator for generating update  
3 instructions to track changes to the multiple target objects and the relationship,  
4 and to update the database when a source object is changed in the database.

1 39. (Original) The manager as claimed in claim 38, wherein the instruction  
2 generator generates the read instructions as SQL Insert statements, SQL Delete  
3 statements or SQL Update statements depending on the change to the source  
4 object.

1 40. (Currently amended) A mapping system for managing an object model  
2 to a relational database, the object model containing object classes having one or  
3 more source objects and target objects having one-to-many relationships of  
4 privately owned type with the source objects, the relational database having tables  
5 including one or more source tables for storing the source objects and one or more

6 target tables for storing the target objects, the source tables having primary keys  
7 and the target tables having foreign keys, the system comprising:  
8 a mapping tool for assisting in mapping object classes to tables and  
9 mapping relationships to foreign keys;  
10 a meta-data storage for storing mapping meta-data defining information of  
11 how object classes map to tables and information of how relationships map to  
12 foreign keys, the mapping meta-data including information of one or more  
13 corresponding target tables for storing the multiple target objects and information  
14 of one or more foreign keys of the corresponding target tables; and  
15 a runtime mapping library for accessing the mapping meta-data in the  
16 meta-data storage and managing object data in the database, the runtime mapping  
17 library having a one-to-many mapping manager for managing one-to-many  
18 relationships and target objects with which a source objects have one-to-many  
19 relationships in accordance with the manipulation of the source object;  
20 wherein by using the foreign key together with the target object data, the  
21 mapping system avoids back references in the target objects.

1 41. (Original) The mapping system as claimed in claim 40, wherein the  
2 one-to-many mapping manager comprises:  
3 an instruction generator for generating instructions to manage the multiple  
4 target objects and the relationship based on the mapping meta-data; and  
5 an instruction executor for executing the instructions on the database and  
6 managing the target objects and the relationship in the database in accordance  
7 with the manipulation of the source object.

1 42. (Original) The manager as claimed in claim 40, wherein the meta-data  
2 storage is provided external to the source object class and the target object classes.

1           43. (Original) The manager as claimed in claim 40, wherein the meta-data  
2 storage is an XML file system for storing the mapping meta-data in XML files.

1           44. (Currently amended) Computer readable media storing the instructions  
2 or statements for use in the execution in a computer of a method for managing  
3 object to relational one-to-many mapping for an object model mapped to a  
4 relational database, the method comprising steps of:  
5           obtaining, for a source object having a primary key value and being  
6 manipulated in a corresponding source table of the relational database, mapping  
7 meta-data including information of a corresponding target table for storing at least  
8 one target object with which the source object has a one-to-many relationship of  
9 privately owned type and information of a foreign key of the corresponding target  
10 table;  
11           generating an instruction to manipulate the at least one target object in the  
12 corresponding target table based on the mapping meta-data; and  
13           manipulating the at least one target object in the database by executing the  
14 instruction on the database, wherein the at least one target object is manipulated  
15 by using the foreign key together with the target object data, thereby avoiding  
16 back references in the target objects.

1           45. (Original) Electronic signals for use in the execution in a computer of a  
2 method for managing object to relational one-to-many mapping for an object  
3 model mapped to a relational database, the method comprising steps of:  
4           obtaining, for a source object having a primary key value and being  
5 manipulated in a corresponding source table of the relational database, mapping  
6 meta-data including information of a corresponding target table for storing at least  
7 one target object with which the source object has a one-to-many relationship of  
8 privately owned type and information of a foreign key of the corresponding target  
9 table;

10           generating an instruction to manipulate the at least one target object in the  
11   corresponding target table based on the mapping meta-data; and  
12           manipulating the at least one target object in the database by executing the  
13   instruction on the database.

1           46. (Original) A computer program product for use in the execution in a  
2   computer of a method for managing object to relational one-to-many mapping for  
3   an object model mapped to a relational database, the computer program product  
4   comprising:

5           a module for obtaining, for a source object having a primary key value and  
6   being manipulated in a corresponding source table of the relational database,  
7   mapping meta-data including information of a corresponding target table for  
8   storing at least one target object with which the source object has a one-to-many  
9   relationship of privately owned and information of a foreign key of the  
10   corresponding target table;

11           a module for generating an instruction to manipulate the at least one target  
12   object in the corresponding target table based on the mapping meta-data; and

13           a module for manipulating the at least one target object in the database by  
14   executing the instruction on the database.